Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

the linkage return address on return from the service routine will raise a program execution exception;

on return from the service routine, attempting to execute the instruction at the linkage return address and raising the chosen exception; and

after servicing the exception, returning control to a caller of the service routine.

REMARKS

The Restriction Requirement of August 26, 2002 proposes to divide the application into Group I (claims 1-32, with claims 82 an 83 now added by amendment), Group II (claims 33-78), and Group III (claims 79-81).

Applicant elects Group I (claims 1-32, 82 and 83), with traverse.

I. Summary of the Argument

Applicant traverses on three separate bases.

First, each Group is misclassified.[1] In each case, the classification appears to be based on a single claim limitation, rather than the claim as a whole. This will not result in an effective or efficient search. For example, a typical claim to an automobile might recite "a body, an engine having certain novel properties, and four wheels with four rubber tires" – but this does not necessarily mean that the claim is appropriately searched in a subclass directed to "rubber compositions." Similarly, when the claims of this application include a single recitation of a "thread scheduler" or "interrupt" as one component in a claim directed to a larger structure or method, this single recitation should not determine the classification of the entire claim group. Rather, each claim – considered as a whole – is clearly directed to subject matter properly searched in class 709, subclass 1.

The title of the application is "Executing Programs For a First Computer Architecture On a Computer Of a Second Architecture," and the particular embodiments described in the specification relate to that overall goal. As described in Section III of the specification (pages 32-51), one use for the invention relates to managing processes or threads in a virtual machine implementation of a CISC machine on a RISC machine. This subject matter is the central focus

[1] A search that focuses on the subclasses designated in the Office Action is unlikely to develop the most-relevant prior art. Incorrect classification directly harms the public interest by reducing the effectiveness and efficiency of examination, in addition to imposing the costs of an inefficient division of the application.

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:23/ST. 17:20/NO. 4862691329 P 7

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

of class 709, subclass 1, directed to "Virtual Machine Task or Process Management," and it appears that the most efficient search for prior art will be in that class and subclass.[2] Because all three groups should be searched in the same class and subclass, there is no "serious search burden," and no basis for division.

Second, the groups recite overlapping subject matter. A search of any of the three groups will all but inevitably result in a search of the subject matter of the other two. There is no "serious search burden" in examining all these claims together.

Third, this disclosure has already been voluntarily divided into twenty-five applications. Because this fine division has already occurred, the claims of this application are directed to patentably-distinct but closely-related aspects of the same invention, and should be examined together.

II. Overview of the Invention

Applicant offers this overview of the specification and figures in order to assist in selecting a more appropriate search class.[3]

The most pertinent portion of the disclosure is Section III, pages 32-51 of the specification, and **Figs. 3a-3n**. "Tapestry" is the name of the system described in the specification. A Tapestry processor has hardware for executing RISC instructions, and several different modes for creating and operating virtual Intel X86 (also known as "IA-32") machines to execute X86 programs. The ability to run programs in either of two instruction sets opens the possibility that a single program might be coded in both instruction sets, and use resources from both architectures.

The Intel X86 has only about 23 registers (depending on how one counts), most of which are only 32 bits wide. An operating system **306** for an X86 computer (such as Microsoft Windows or IBM OS/2) only has software capabilities for managing these 23 on a context switch. In contrast, referring to **Fig. 3a** and to Table 1 at page 21-22 of the specification, the

---

[2] It will be understood that this introductory discussion of the invention and the claims (and analogous introductions to the other claims made in this paper) is an indication of the general search field most likely to be relevant to the claim, not a statement of the scope of the claims.

[3] One specific embodiment of the invention is discussed here, in order to provide context to assist the examiner's search. This discussion is directed to assisting in framing an efficient search, and therefore is intended to direct the Examiner to the place where the best prior art is likely to be. It should be understood that this contextual discussion of one particular embodiment is not a limiting discussion of the invention or the claims.

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:23/ST. 17:20/NO. 4862691329 P 8

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

Tapestry processor has 64 general registers, named "r0" through "r63," which are 64 bits wide. An unaltered, unassisted X86 operating system cannot context switch a 64-register process.

In the Tapestry System, X86 threads (*e.g.*, **302, 304**) carry the normal X86 context, including the X86 registers, as represented in the low-order halves of r32-r55, the EFLAGS bits that affect execution of X86 instructions, the current segment registers, etc. In addition, an X86 thread **302, 304** may embody a good deal of "extended context" (context state that is stored in the portion of the Tapestry processor context beyond the content of the X86 architecture), including the various Tapestry processor registers, general registers r1-r31 and r56-r63, and the high-order halves of r32-r55 (see Table 1), the current value of ISA bit **194** (a bit that tells whether the hardware is currently executing in native Tapestry RISC mode or X86 mode), and other processor status and control information.

An unaltered, unassisted off-the-shelf X86 operating system **306** cannot manage threads **302, 304** that rely on extended context. Because X86 operating system **306** is coded in the X86 instruction set, and the X86 instruction set does not have instructions to access the extended context resources, the extended context cannot even be read or written by X86 operating system **306**, let alone switched on a context switch.

The Tapestry system performs some additional housekeeping on entry and exit to virtual X86 **310**, in order to save and restore the extended context, and to maintain the association between extended context information and threads **302, 304** managed by X86 operating system **306**.

Figs. 3a-3n describe one mechanism that can be used to save and restore the full context of an X86 thread **304** that is currently using Tapestry extended resources. In overview, this mechanism snapshots the full extended context into a memory location **355** that is architecturally invisible to virtual X86 **310**. A correspondence between the stored context memory location **355** and its X86 thread **304** is maintained by Tapestry virtualization system **312, 316** in a manner that that does not require cooperation of X86 operating system **306**, so that the extended context will be restored when X86 operating system **306** resumes X86 thread **304**, even if X86 operating system **306** performs several context switches among X86 threads **302** before the interrupted X86 thread **304** resumes. The Tapestry virtualization system **312, 316** briefly gains control at each transition from X86 to Tapestry or back, including entries to and returns from X86 operating system **306**, to save the extended context and restore it at the appropriate time.

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28. 02 17:24/ST. 17:20/NO. 4862691329 P 9

Received from < 2124593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

The details of that context saving and restoring are described in further detail in the specification, and in **Figs. 3i, 3j**, and **3k**. When a process is about to enter X86 operating system **306**, a save slot **355** (a block of storage sized to hold a full Tapestry context snapshot) is allocated **361** from among those currently free. The entire Tapestry context, including the X86 context and the extended context, is saved **362** into the context space **356** of allocated save slot **355**. The process context is then overwritten with information that enables the Tapestry virtualization system **312** to find the save slot – in other words, X86 operating system **306** never actually sees the context of the process that it is scheduling, only unintelligible "junk" cobbled up by Tapestry virtualization system **306**. X86 operating system **306** is then invoked to handle the interrupt.

Referring to **Fig. 3h**, section **370** shows resumption of a process after the X86 operating system **306** completes. The contents of the current process is examined to determine which process is has been resumed by the X86 operating system. Of course, that process is not ready to execute –the context contains the Tapestry-specific "junk." Instead, Tapestry virtualization system **312, 316** restores the correct context from the save slot, and then the process is allowed to resume.

III. All Three Claim Groups are More Appropriately Searched in Class 709, Subclass 1

A. Group I (claims 1-32, 82 and 83) is more appropriately searched in class 709, subclass 1

The Restriction Requirement proposes to search Group I (claims 1-32, 83 and 83) in class 709, subclass 102. Claim 1, a representative claim from Group I, recites as follows (paragraph numbering added):

> 1. A method, comprising:
>
> [1] without modifying a pre-existing operating system of the computer, establishing an entry exception to be raised on each entry to the operating system at a specified entry point or on a specified condition, the entry exception having an associated entry handler, the entry handler programmed to save a context of an interrupted thread and modify the thread context before delivering the modified context to the operating system;
>
> [2] without modifying the operating system, establishing a resumption exception to be raised on each resumption from the operating system complementary to one of the specified entries, the resumption exception having an associated exit handler, the exit handler programmed to restore the context saved by a corresponding execution of the entry handler.

FROM SCHULTE ROTH & ZABELLLP (MON) 10. 28' 02 17:25/ST. 17:20/NO. 4862691329 P 10

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

[3] scheduling concurrent threads of control by the operating system, each thread having an associated context, the association between a thread and a set of computer resources of the associated context being maintained by the operating system;

[4] on detecting a specified entry to the operating system from an interrupted thread of the computer, raising and servicing the entry exception; and

[5] on detecting a complementary resumption, raising and servicing the resumption exception, and returning control to the interrupted thread;

[6] the entry exception, exit exception, entry handler, and exit handler being cooperatively designed to maintain an association between a one of the threads and an extended context of the thread through a context change induced by the operating system, the extended context including resources of the computer associated with the thread beyond those resources whose association with the thread is maintained by the operating system.

In pertinent part, the class definition for class 709, subclass 102 reads as follows:

CLASS 709: ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: MULTIPLE COMPUTER OR PROCESS COORDINATING

SECTION I - CLASS DEFINITION

GENERAL STATEMENT OF THE CLASS SUBJECT MATTER

This class provides for an electrical computer or digital data processing system or corresponding data processing method including method or apparatus for transferring data or instruction information between a plurality of computers or processes wherein the computers or processes employ the data or instructions before or after transferring and the employing affects said transfer of data or instruction information. The class includes the following subject matter:

A. Process or apparatus for administrating process or job execution over a digital data processing system.

B. Process or apparatus for transferring data among a plurality of spatially distributed (i.e. situated, at plural locations) computers or digital data processing systems via one or more communications media (e.g., computer networks).

C. Process or apparatus for exchanging data or messages between two executing programs or processes, generally independent of the hardware used in the communication.

D. Process or apparatus for synchronizing control or regulation of clocking or timing operations of two or more processors.

SCOPE OF THE CLASS

This class is limited to digital data processing systems and functions for transferring unspecified data or instruction information and the processing thereof by digital data processing systems. Systems concerned with movement or processing of other specific types of

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:25/ST. 17:20/NO. 4862691329 P 11

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

information and digital signals, per se, are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

100 TASK MANAGEMENT OR CONTROL:

This subclass is indented under the class definition. Subject matter comprising means or steps for administrating over processor or job execution in a digital data processing system.

102 PROCESS SCHEDULING:

This subclass is indented under subclass 100. Subject matter comprising means or steps for scheduling multiple tasks based upon any considered factors, e.g., priority of execution, balancing the work load or resources, memory use, register use, resource availability, time constraints, etc.

(1) Note. Included here is task assignment, (i.e., deciding which processor or other resources will be used to execute one or more tasks).

(2) Note. Signaling, semaphores, and mutual exclusion mechanisms (i.e., mutexes) used for program or process synchronization purposes are classified here. ...

Claim 1 recites "scheduling concurrent threads" only in paragraph [3] of claim 1. The five other paragraphs of the claim make no further reference to "process scheduling," and recite many features not encompassed within a search class directed to "process scheduling." A search that is confined to only paragraph 3 of the claim will not be thorough, efficient, or effective.

Rather, the embodiment described in the specification[4] clearly relates to class 709, subclass 1. Indeed, the title of the application nearly matches the class definition. The pertinent portion of subclass 1 reads as follows:

1 VIRTUAL MACHINE TASK OR PROCESS MANAGEMENT:

This subclass is indented under the class definition. Subject matter comprising means or steps operating on a computer or digital data processing system which enable a first type of processor to emulate and execute instructions associated with one or more different types of processors.

(1) Note. This subclass is directed to subject matter encompassing one or more virtual machines that execute in single task, or multitasking, operating system environments.

---

[4] This is only the embodiment in the specification, not the scope of the claims. The claims may well be practiced in contexts outside class 709, subclass 1, and applicant has no objection to searching other potential subclasses, but that does not change the fact that the most efficient and effective search will be the search that focuses on the embodiment that is the subject of many of the dependent claims. The primary search – and the classification for restriction purposes – should be the subclass most related to the embodiments in the disclosure, and claimed in the dependent claims.

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:26/ST. 17:20/NO. 4862691329 P 12

Received from < 2124593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

(2) Note This subclass includes computers or digital data processing systems executing a plurality of virtual machines that are preemptively or nonpreemptively scheduled.

(3) Note. This subclass includes means or steps for mimicking the performance of one processing device within another processing device. For example, a software program that allows applications written for a first computer to be executed on a different second computer interpreting the machine instructions for the first computer, thereby becoming a virtual machine.

The title, the specification (*see, e.g.*, specification, page 26, line 8; page 27, line 10; page 33, lines 3, 8, 9, and 18; page 40, lines 10, 20 and 23, all showing that the specification is directed to implementation of a "virtual machine" or "virtual X86"), and a number of the dependent claims (*see, e.g.*, claims 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 16, 20, 21, 22, 24) clarify that prior art with respect to the entirety of these claims (rather than isolated parts), if it exists at all, is much more likely to be found in subclass 1 than subclass 102. A search of subclass 102 has a significant risk of being entirely ineffective with respect to the majority of the dependent claims.

For these reasons, Applicant suggests that Group I should be searched in class 709, subclass 1 instead of subclass 102. Subclass 102 may well turn out to be an appropriate class into which to expand a secondary search, but it cannot be the best primary search class.

B. Group II (claims 33-78): the classification proposed in the Restriction Requirement is clearly incorrect

The Restriction Requirement proposes to search claims 33-78 in class 710, subclass 260. This is clearly incorrect. Group II should be searched with Group I, whether Group I is searched in subclass 1 or subclass 102.

Claim 56, a representative claim from Group II, recites as follows:

> 56. A method, comprising:
>
> without modifying a pre-existing operating system of the computer, establishing an entry handler for execution at a specified entry point or on a specified entry condition to the operating system, the entry handler programmed to save a context of an interrupted thread and modify the thread context before delivering the modified context to the operating system;
>
> without modifying the operating system, establishing an exit handler for execution on resumption from the operating system following an entry through the entry handler, the exit handler programmed to restore the context saved by a corresponding execution of the entry handler.

In relevant part, the definition for class 710 reads as follows:

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:26/ST. 17:20/NO. 4862691329 P 13

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

CLASS 710: ELECTRICAL COMPUTERS AND DIGITAL DATA PROCESSING SYSTEMS: INPUT/OUTPUT

SECTION I - CLASS DEFINITION

This class provides, within a computer or digital data processing system, for the following subject matter:

A. Processes or apparatus for transferring data from one or more peripherals to one or more computers ...;

B. Processes or apparatus for interconnecting or communicating between two or more components connected to an interconnection medium (e.g., a bus) within a single computer or digital data processing system;

(2) Note. Classification herein requires more than nominal recitation of "peripheral devices," "peripherals," "input/output," or "I/O," or of intrasystem connections or communications.

As is readily observed, claim 56 has nothing whatsoever to do with "input/output." Claim 56 contains no recitation whatsoever of "'peripheral devices,' 'peripherals,' 'input/output,' or 'I/O,' or of 'intrasystem connections or communications,'" not even "nominal recitation."[5] Note (2) to class 710 requires that claim 56 be "classified elsewhere."

Nor is claim 56 a good match for subclass 260. The definition for subclass 260 reads as follows:

260 INTERRUPT PROCESSING:

This subclass is indented under the class definition. Subject matter comprising means or steps for stopping, halting, or suspending a current processing function within a computer or digital data processing system.

Claim 56 does not recite "means or steps for stopping, halting, or suspending a current processing function." Rather, considered as a whole, claim 56 recites a mechanism invoked on entry to or exit from an operating system, typically when some event outside the claim raises an interrupt, used as a means for "saving" and "restoring" thread context, the subject matter of class 709, subclass 1. The word "interrupt" appears nowhere in claim 56.

In any event, subclass 260 is clearly a misclassification: a search of subclass 260 will be an inefficient use of the Examiner's time.

In the embodiment described in the specification[6], the invention of claim 56 is used as a mechanism for managing processes in a virtual machine emulation system. (*See, e.g.*, specification, page 26, line 8; page 27, line 10; page 33, lines 3, 8, 9, and 18). As becomes

---

[5] In some cases, the process of claim 56 could be performed incidentally to an input/output operation. However, incidental occurrences do not form the best basis for choosing a search class.

[6] See footnote 4.

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:27/ST. 17:20/NO. 4862691329 P 14

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

clearer by consideration of the dependent claims (*e.g.*, claims 35, 36, 37, 38, 40, 41, 42, 44) Group II should be searched as "virtual machine task or process management," class 709, subclass 1. Class 709, subclass 102, may turn out to be a useful secondary search class, confirming that Group II should be searched and examined with Group I.

C. Group III (claim 79-81): the classification proposed in the Restriction Requirement is clearly incorrect

The Restriction Requirement proposes to classify Group III in class 709, subclass 318. This is clearly incorrect. Claim 79 recites as follows:

> 79. A method, comprising:
>
> during invocation of a service routine of a computer, passing a linkage return address to the service routine at which to resume execution on completion of the service, the linkage return address being deliberately chosen so that an attempt to execute an instruction from the linkage return address on return from the service routine will raise a program execution exception;
>
> on return from the service routine, attempting to execute the instruction at the linkage return address and raising the chosen exception; and
>
> after servicing the exception, returning control to a caller of the service routine.

In pertinent part, the definition for subclass 318 reads as follows:

> **310 INTERPROGRAM COMMUNICATION, INTERPROCESS COMMUNICATION (IPC):**
>
> This subclass is indented under the class definition. Subject matter comprising means or steps for exchanging data or messages between two executing programs or processes, independent of the hardware used in the communication.
>
> (2) Note. The subject matter of this subclass is directed to communication between processes.
>
> (6) Note. This subclass is for communication between processes and tasks, communication between computers or digital data processing systems and peripherals is classified elsewhere.
>
> **318 EVENT HANDLING OR EVENT NOTIFICATION:**
>
> This subclass is indented under subclass 310. ...

To fall within subclass 310, a claim must recite communication between "two executing programs or processes." However, claim 79 is practiced when one single process invokes a

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:27/ST. 17:20/NO. 4862691329 P 15

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

service routine. There is no recitation of "two processes."[7] Claim 79 cannot possibly fall within subclass 310, and therefore cannot fall within subclass 318.

As with Groups I and II, the embodiment described in the specification[8] relates to managing processes in a virtual machine emulation system (see pages 49-50 of the specification). Class 709, subclass 1 is a more appropriate focus for search, with subclass 102 being a reasonable subclass into which to expand the search.

D. Because all three Groups should be searched together, there is no "serious search burden," and thus no ground for Restriction

Applicant traverses the Requirement for Restriction between groups I and II. Between Groups I and II, the facts as stated by the Examiner do not meet the legal requirements for a proper Restriction.

MPEP § 803 states the requirements for a restriction requirement (emphasis added):

> There are <u>two</u> criteria for a proper requirement for restriction between patentably distinct inventions:
> (1) The inventions must be independent (see MPEP §802.01, §806.04, §808.01) or distinct as claimed (see MPEP §806.05 - § 806.05(i)) <u>and</u>
> (2) There must be a <u>serious</u> burden on the examiner if restriction is not required (see MPEP §803.02, §806.04(a)-(j), §808.01(a) and §808.02).

MPEP § 803 clarifies (emphasis added):

> If the search and examination of an entire application can be made without <u>serious burden</u>, the examiner must examine it on the merits, even though it includes claims to distinct or independent inventions.

1. The August Restriction Requirement is incomplete

Applicant concedes that Groups I, II and III are distinct (criterion (1) of MPEP § 803), but traverses under criterion (2). The Restriction Requirement omits any mention of criterion (2) — the Restriction raises no showing of a serious search burden. At a minimum, a Restriction Requirement without such a showing is incomplete, and cannot be made final.

---

[7] Claim 79 could, conceivably, also be practiced by two processes, to the degree no limitation excludes the possibility. However, theoretical possibilities are not proper bases on which to classify claims for primary search.

[8] See footnote 4.

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:28/ST. 17:20/NO. 4862691329 P 16

Received from < 2124593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

2. As correctly classified for search, the three Groups are properly examined together

The Examiner has conceded that Groups I, II and III are "related" (paragraph 5). "Related" groups may not be restricted when they are classified together for search. MPEP § 808.02 ("Where, however, the classification is the same and the field of search is the same and there is no clear indication of separate future classification and field of search, no reasons exist for dividing among related inventions."). Because all claims should be searched in the same search class and subclass, restriction is inappropriate.

IV. A Search of Group I Will Inevitably Involve Searches of Groups II and III

Any search burden is unlikely to be "serious" when claims 33 and 56 (the independent claims of Group II) appear nearly word-for-word in claims 1, 6 and 82 of Group I. Claim 1 must be searched in any event, and claims 6 and 82 must be searched unless claim 5 is given a first action allowance. Therefore, it seems that a search of Group I will necessarily result in a search of the independent claims of Group II as well.

Similarly, a search of Group I will involve a search of claim 83, and a search of Group II will involve a search of claim 52. These claims recite subject matter that, at least for search purposes, closely overlaps the subject matter of claims 79-81 (the claims of Group III).

Thus, a search of Group I (the group elected) will all but inevitably involve a search of the other Groups. Because no "serious" search burden exists, the claims are appropriately examined together.

V. Further Facts Suggest Restriction is Inappropriate

This disclosure has been voluntarily divided into 25 applications. The 24 other applications are listed in the Information Disclosure Statement filed January 17, 2001. Because of this fine division, it should be apparent that the claims in this application are closely related, and that further division is unwarranted.

Further, by the time this application receives its first action on the merits, the application will have been pending for four years. 35 U.S.C. § 154(b)(1)(A)(i) indicates Congressional intent that an application receive its first consideration within fourteen months of filing, and that prosecution conclude within three years. If this restriction is affirmed, the divided claims may well not receive their first consideration for many more months. Even now, these delays are far in excess of the statutorily-prescribed guideline for PTO action. *Telecommunications Research*

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:28/ST. 17:20/NO. 4862691329 P 17

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

and Action Center v. Federal Communications Commission, 750 F.2d 70, 80 (D.C. Cir. 1984) ("The time agencies take to make decisions must be governed by a 'rule of reason,' [and] where Congress has provided a timetable or other indication of the speed with which it expects the agency to proceed ..., that statutory scheme may supply content for this rule of reason.") Failure to act within this time is "unreasonable delay" in violation of the Administrative Procedure Act, 5 U.S.C. § 555(b). *Id.* Imposing further delay by restricting claims would not be proper.

If division of this application was ever proper, the time for doing so is long past. In view of the undue delay on the part of the Office, far in excess of statutory guideline, and substantial loss of patent term occasioned thereby, it would be entirely inappropriate for the PTO to now divide the application on anything less than the clearest showing of proper division of the application. If grounds for division validly exist at all, they are tenuous at best. No division should be required.

Applicant requests that the application be passed to issue in due course. The Examiner is urged to telephone Applicant's undersigned counsel at the number noted below if it will advance the prosecution of this application, or with any suggestion to resolve any condition that would impede allowance. Enclosed is a Petition for Extension of Time for one month. In the event that further extension of time is required, Applicant petitions for that extension of time required to make this response timely. Kindly charge any additional fee, or credit any surplus, to Deposit Account 50-0675, Order No. 5231.5-4013.

Respectfully submitted,
SCHULTE ROTH & ZABEL

Dated: October 28, 2002

By: _____
David E. Boundy
Registration No. 36,461

Mailing Address:
SCHULTE ROTH & ZABEL
919 Third Avenue
New York, New York 10022
(212) 756-2000
(212) 593-5955 Telecopier

FROM SCHULTE ROTH & ZABELLLP (MON) 10. 28' 02 17:29/ST. 17:20/NO. 4862691329 P 18

Received from < 212+593 > at 10/28/02 5:29:14 PM [Eastern Standard Time]

REWRITTEN CLAIMS MARKED UP TO SHOW CHANGES

82. (new) The method of claim 5, further comprising the step of:

without modifying a pre-existing thread scheduler of the computer, establishing an entry handler for execution at a specified entry point or on a specified entry condition to the thread scheduler, the entry handler programmed to save a context of an interrupted thread and modify the thread context before delivering the modified context to the thread scheduler.

83. (new) The method of claim 20, further comprising the steps of:

during invocation of a service routine of the operating system, passing a linkage return address to the service routine at which to resume execution on completion of the service, the linkage return address being deliberately chosen so that an attempt to execute an instruction from the linkage return address on return from the service routine will raise a program execution exception;

on return from the service routine, attempting to execute the instruction at the linkage return address and raising the chosen exception; and

after servicing the exception, returning control to a caller of the service routine.

FROM SCHULTE ROTH & ZABELLLP. (MON) 10. 28' 02 17:29/ST. 17:20/NO. 4862691329 P 19